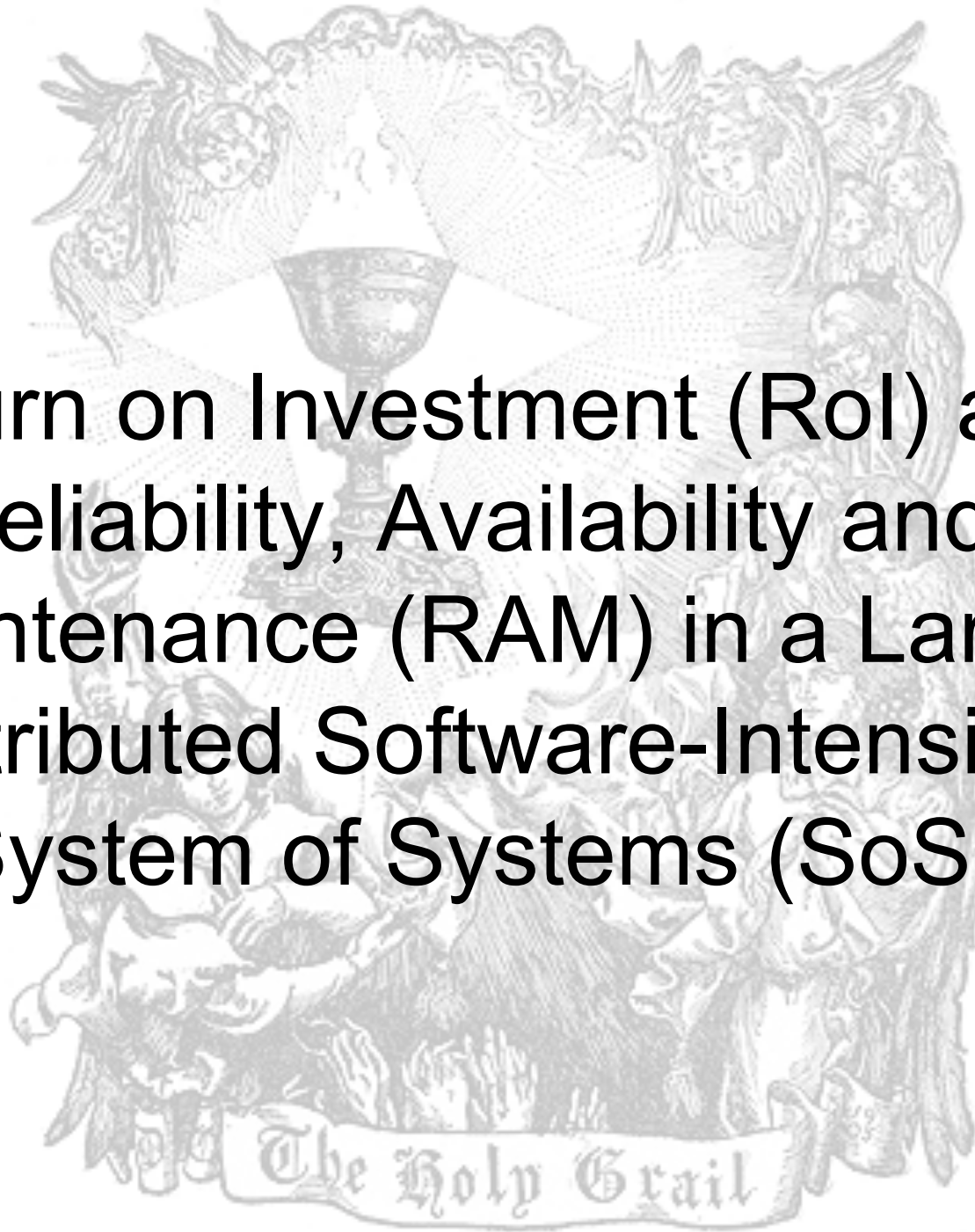
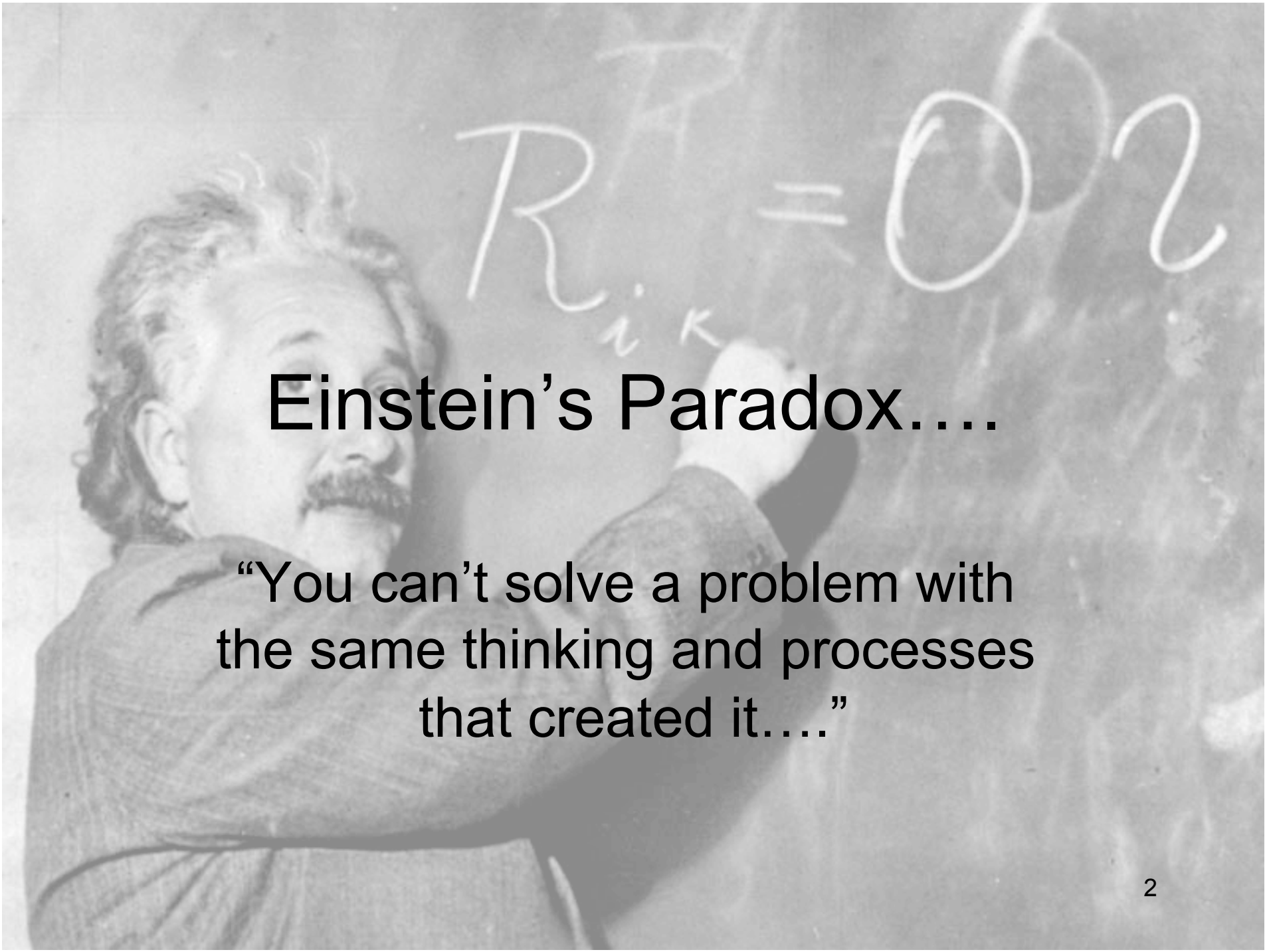


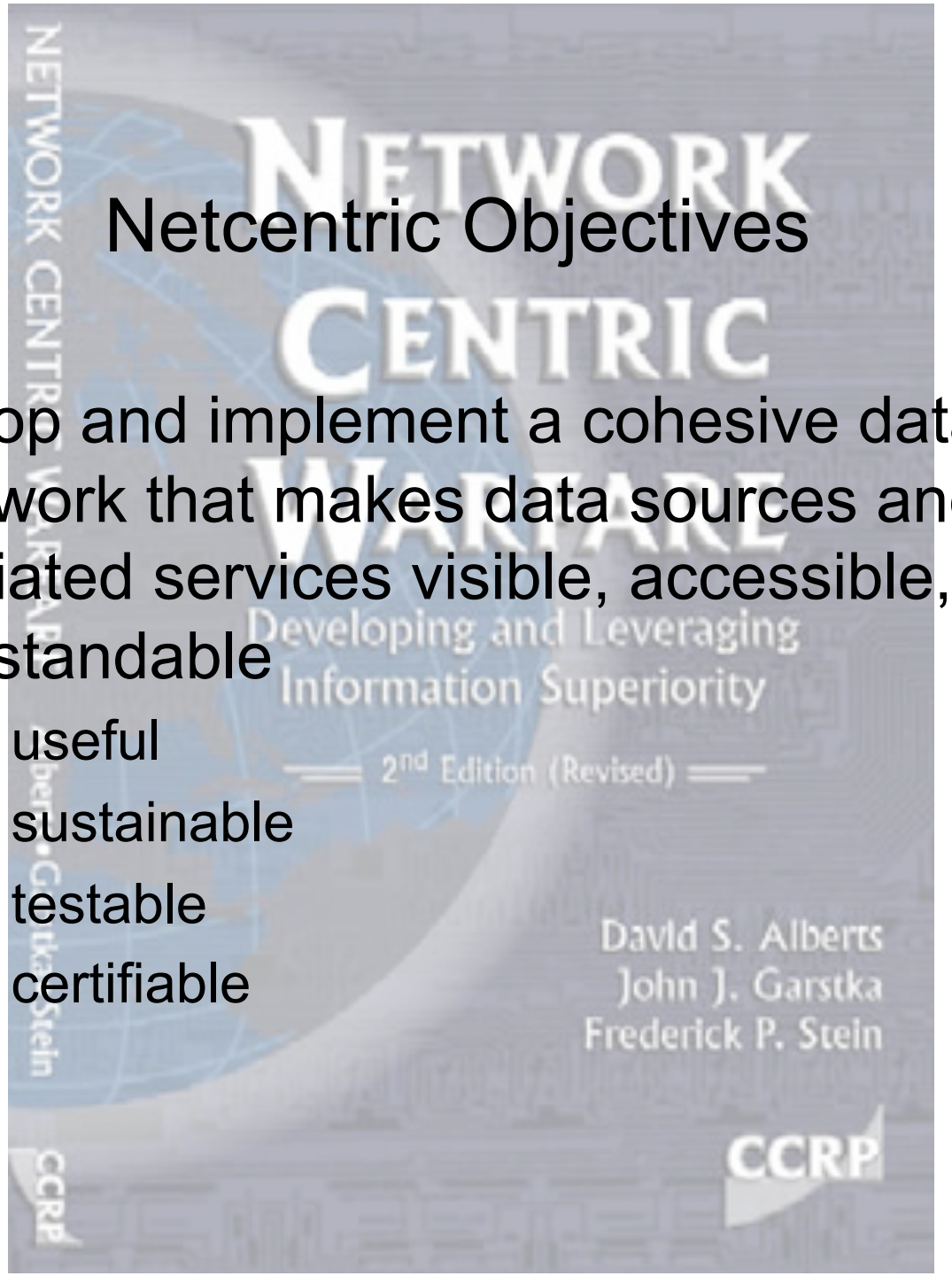
Return on Investment (RoI) and Reliability, Availability and Maintenance (RAM) in a Large Distributed Software-Intensive System of Systems (SoS)



Einstein's Paradox….

"You can't solve a problem with the same thinking and processes that created it…."

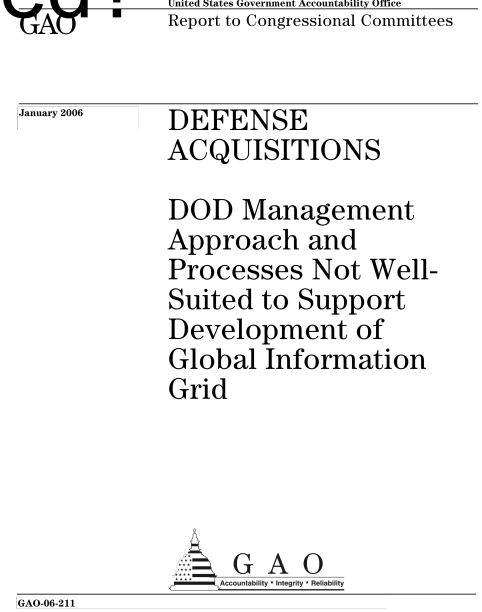
Netcentric Objectives



- Develop and implement a cohesive data framework that makes data sources and associated services visible, accessible, and understandable
  - ✓ And useful
  - ✓ And sustainable
  - ✓ And testable
  - ✓ And certifiable

Why will some program finally succeed where others have failed?

- GAO reports* re
  - GIG
  - NMCI
  - FCS
  - JTRS
  - Etc
- Defense Science Board (DSB) FY 09 reports* re
  - IT Acquisition
  - Urgent Operational Needs



*Netcentric programs are all behind schedule and over budget…IT acquisition process is broken...

Netcentric ROI Objective: ***Better-***Speed-to-***Better***-Capability

Measurably and testably:

- Deploy capability faster
- Improve operational outcomes
- Improve delivered-capability-per-cost ratio
- Improve predictability of cost and time per delivered capability

Through:

- Rapid, incremental, parallel, D, T&E and C&A
- Reusing components in build-time and run-time
- Creative contracting

Netcentric System Long Poles*

- Information Assurance (IA)
  - Legacy C&A does not support netcentricity
  - Security models and products for SOA immature
- Semantic Interoperability (SI)
  - Data glut leads to needle-in-a-haystack issue
  - Semantic technology state-of-the-art inadequate

*Must address both issues up front, realistically, and in context with each other …. Both policy & technology!

Threat vs. Opportunity

- Difficult specialized tasks
- Governing directives are vague
- Cross POR collaboration is usually problematic

Threat = Typically bureaucratic, “compliant”, PowerPoint-engineering solution (per GAO and DSB reports)

Threat vs. Opportunity

<u>Opportunity</u> = Perform genuine, lead-from-the-front, netcentric engineering

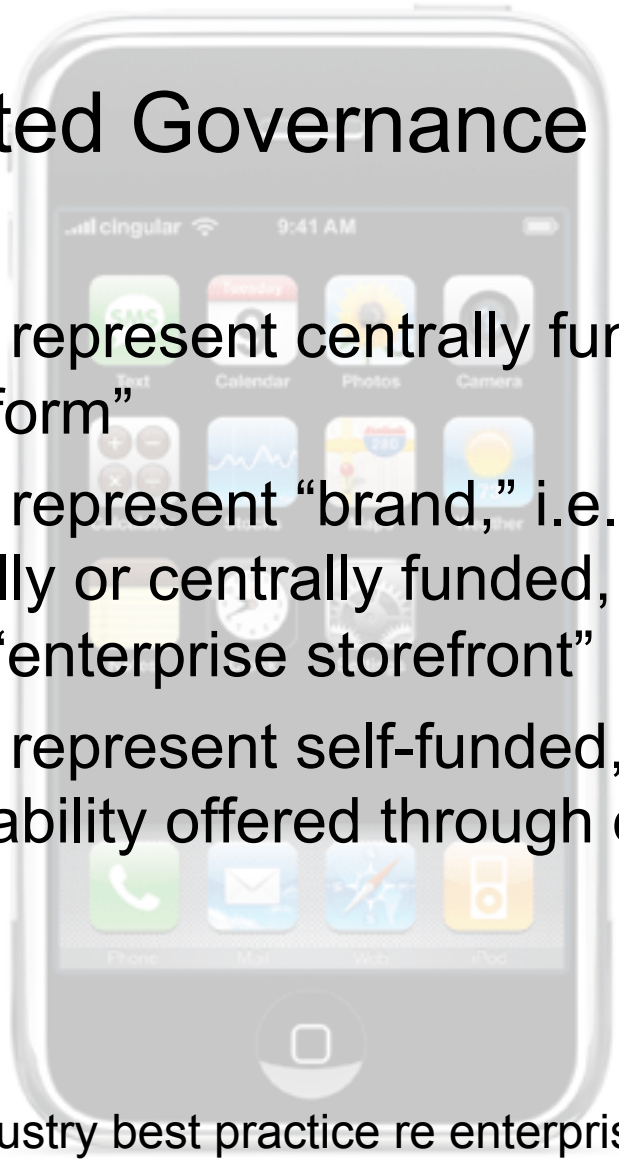
- Leverage existing success cases
- Distill an objective framework from vague policy
- Develop realistic enterprise value proposition and “sell” to AOC community stakeholders

Netcentric Value Proposition

- Governing directives* mandate “Risk Adaptive Access Control” (RAdAC), NR-KPP, reciprocal C&A, and Sustainability-KPP (S-KPP)
- Guidance is vague, leaves implementation detail to programs.
- We can develop value-based detail:
  - S-KPP = speed-to-capability = low cost, certified, off-the-shelf components
  - NR-KPP = testable, link between operational MOE and Information Processing Efficiency (IPE).
    - IA is a critical component of NR-KPP
      - RAdAC is the GIG-mandated IA model
- RAdAC **NR-KPP RoI demonstrated in CWID 08** (High Assurance Tactical SOA (HATS) pilot series)
  - **20% improvement in probability of detection of High Value Target**
  - **100% improvement in detect-to-engage time**
- RAdAC **S-KPP RoI is equivalent to COTS SOA RoI**
  - SOA re-usable components **cost ~20% more up front** than non-reusable coding
  - **Speed-to-capability first article = ~1 year vs. ~ 6 years** for traditional acquisition
  - SOA => **2.5 X more re-use** than traditional models.
  - Enterprise re-use results in **90% cost reduction** over new development.
    - Sample case = Integrity-as-a-service **30K lines of code @ $5k/line X 1 as a service vs. X many** as a traditional capability.; Re-useable high assurance components **decreases time and cost for C&A**
- **Achieving RoI depends on expert architect/engineer executing new enterprise paradigm!**

*NSA GIG IA Arch.; CJCSM 3170; CJCS 6212; ICD 503; DoDI 5000/8500

Federated Governance Model*

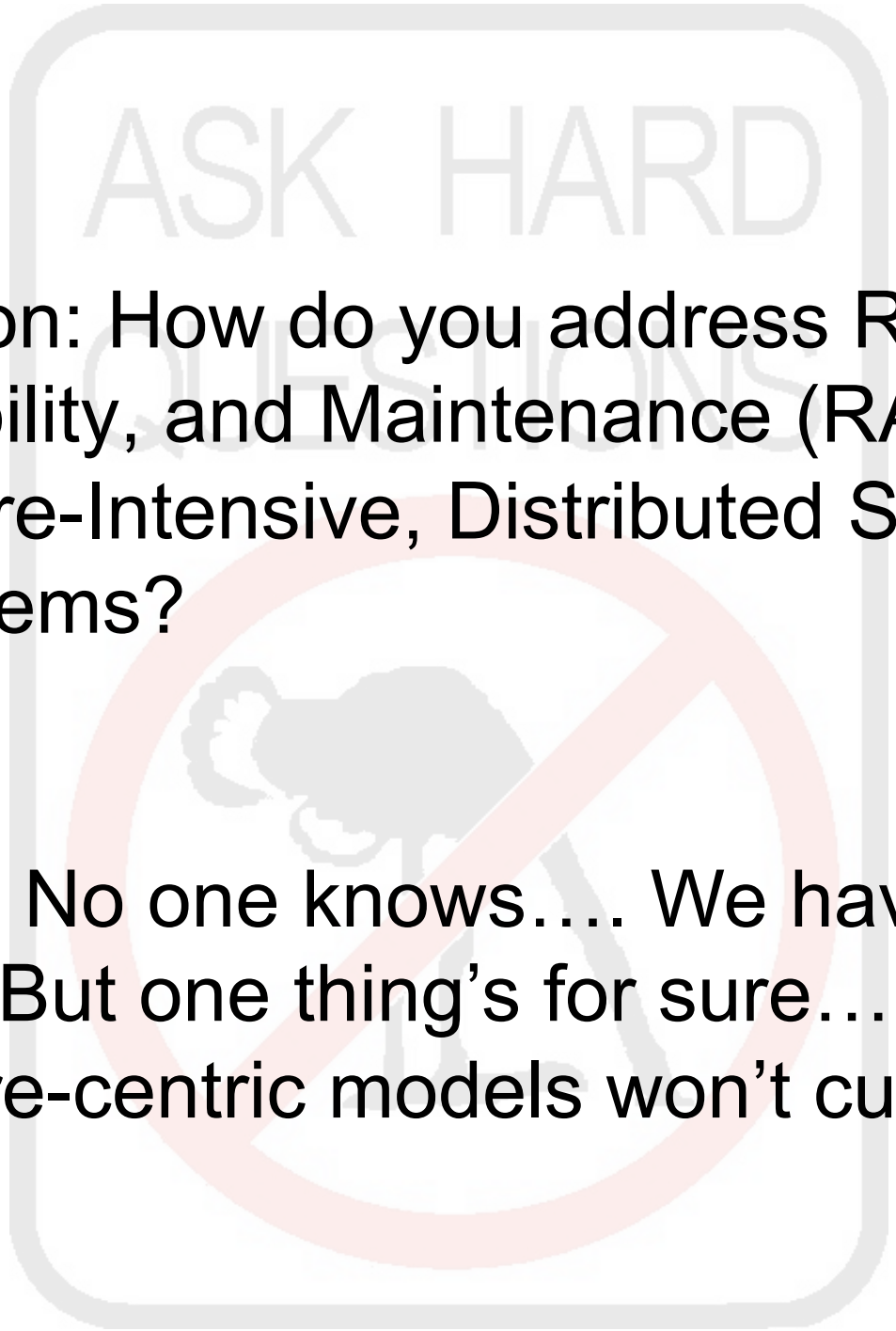


- Tier 0 services represent centrally funded, and managed “platform”
- Tier 1 services represent “brand,” i.e. locally managed, locally or centrally funded, verifiably interoperable, “enterprise storefront”
- Tier 2 services represent self-funded, independent, innovative capability offered through enterprise storefronts.

*Per industry best practice re enterprise SOA, e.g. iPhone, e-Bay developers, Google gadgets, e-File, etc.

Netcentric Platform Model

- Establish GIG business model = e-Portal for consumable *off-the-shelf* (OTS) = COTS, GOTS & Open Source Software (OSS) *certified* net-ready components
- Define generic and objective net-ready assessment categories and methods (not universal specifications!) per enterprise business objectives
- Use a net-ready “logo” to create a federation of qualified, motivated, independent government, industry, and academic net-ready providers
- Base acquisition on components that can reduce risk re: cost, performance, and schedule and *deliver capability faster*.
  - Require logo as “responsive” to GIG procurements
  - Bake evolutionary COTS process into FAR boilerplate
  - Hardwire cross program collaborative work flow



Question: How do you address Reliability, Availability, and Maintenance (RAM) in a Software-Intensive, Distributed System-of-Systems?

Answer: No one knows…. We have to figure it out.... But one thing's for sure… the old hardware-centric models won't cut it.

Notional Example
"C2 Program X" Reliability Availability & Maintenance (RAM) ROI

Demonstration of how relationship between A_{nr} and A_o improves Reliability Maintenance and Availability (RMA) while reducing cost

Assumptions

$$A_o = \frac{MTBF}{MTBF + MTTR + MLDT} = 0.99999$$

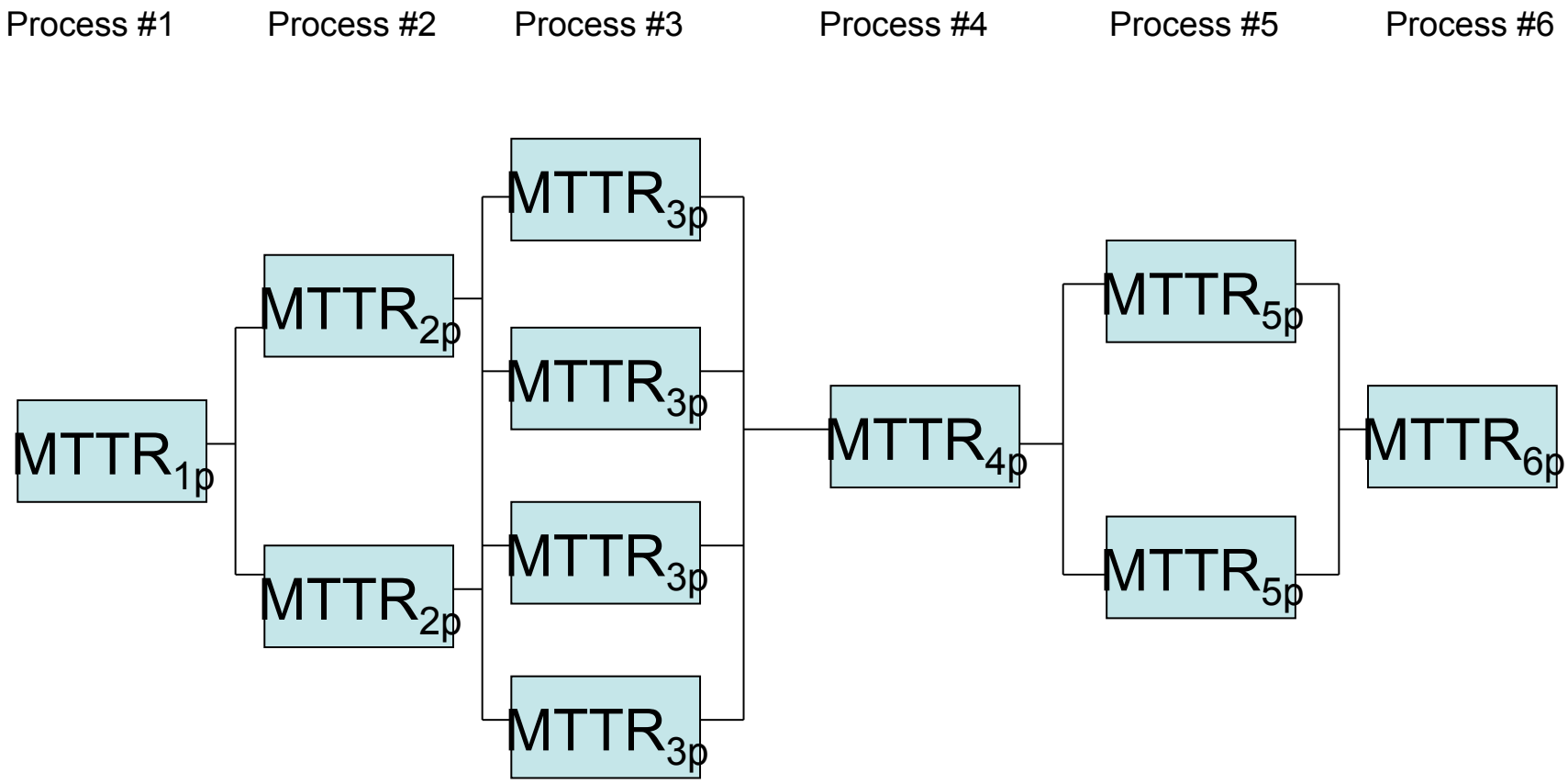
and

$$A_{nr} = \frac{(T_D)_{ie}}{(T_D + T_T + T_C)_{ce}} = 0.66$$

A_o = Operational Availability
A_{nr}= Net-ready Availability
T_D = Development Time
T_T = (Additional) Test Time
T_C = (Additional Certification Time)
$()_{ie}$ = Initial Estimate
$()_{ce}$ = Current Estimate

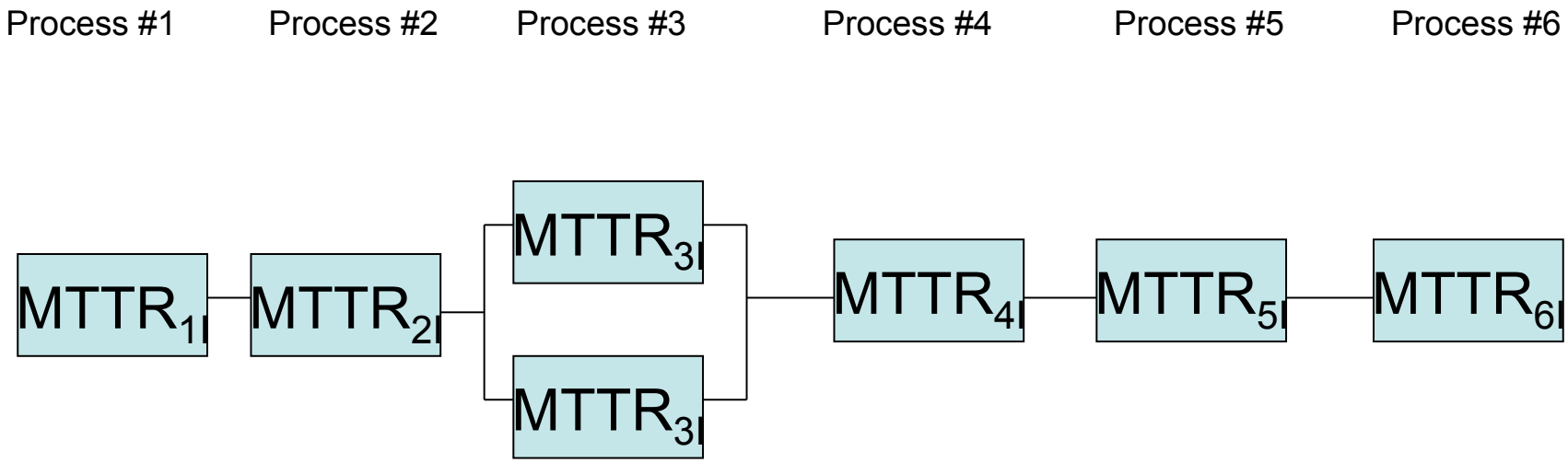
where $(T_D)_{ie} <= (T_D)_{ce}$ and $(T_D + T_T + T_C)_{ce} <= 18$ mos
(to simplify notional example)

At PDR, MTTR is xxxx hrs given:



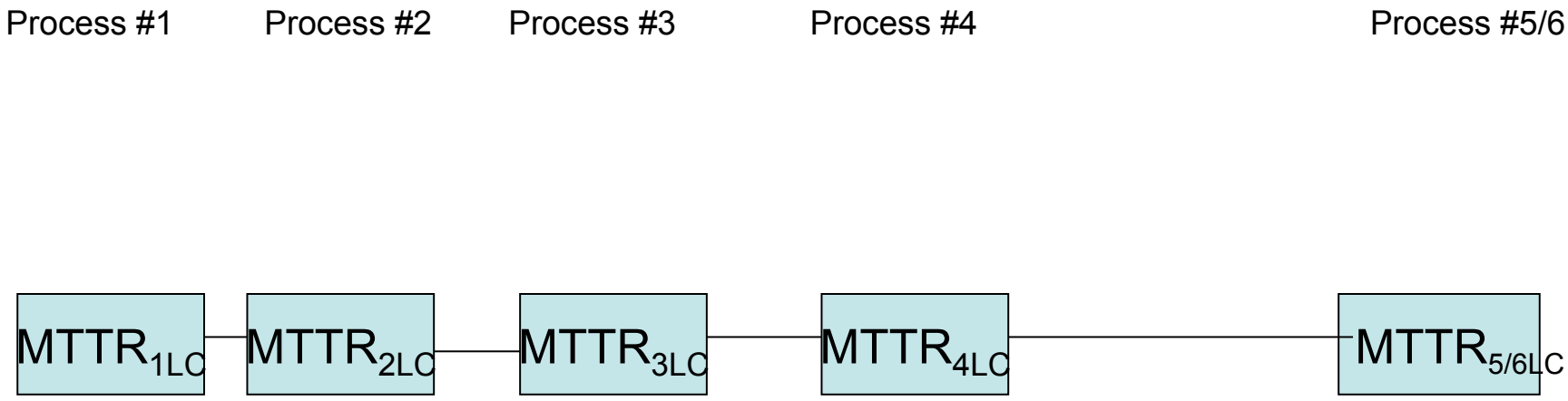
A_o = 0.99999, A_{nr} = 0.66, Cost = $$$$$$

At IOC, (IOC = PDR + 18months) MTTR is xxxx hrs, given:



Where, $MTTR_{1I} < MTTR_{1P}$, $MTTR_{2I} < MTTR_{2P}$, etc., and $A_o = 0.99999$, $A_{nr} = 0.66$, Cost = $$$$, or…
$A_o = 0.9XXXX$, $A_{nr} = 0.66$, Cost = $$

At Lifecycle Support Contract Performance Review (IOC + 18months) MTTR is xxxx hrs, given:



Where, $MTTR_{1LC} < MTTR_{1I}$, $MTTR_{2LC} < MTTR_{2I}$, etc., over each improvement cycle and, $A_o = 0.99999$, $A_{nr} = 0.66$, Cost = $$$$ …. or Ao = 0.99XXX, Anr = 0.66, Cost = $$ ….

Draft Executive Brief for C2 Program X Management

Reliability, Availability, and Maintenance Issue

- Program X aims to deliver C2 capability via new “open system” paradigm: leverage SOA and COTS to deliver continuous evolutionary improvement.
- Program X inherited KPP, Ao=0.99999, designed for legacy “closed system” paradigm.
- Issue is that Program X aims to provide software-enabled “Information Availability”, but traditional Ao is a H/W-centric metric.

SOA Program
Scoring Conference (SC)

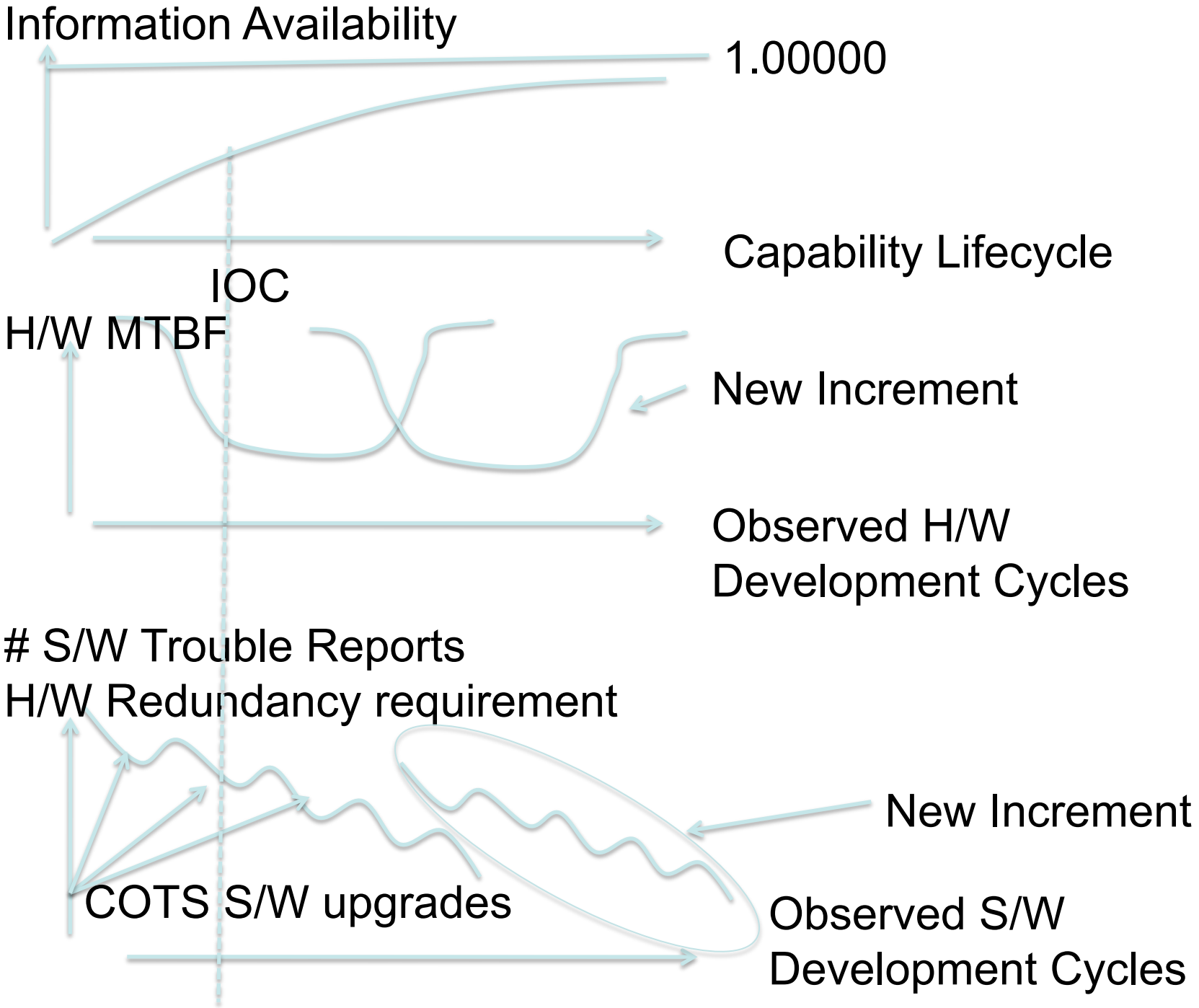
- Typical SC recognizes the issue:
  - Uses Business Process Modeling (BPM) to define Fully Mission Capable (FMC) and “Available, Degraded” (Deg)
  - Considers both h/w & s/w
- However, SC analysis finds most failures as h/w issues because:
  - H/w failures are material and s/w failures are not.
  - H/w fails after thousands of hours, s/w fails in tens or hundreds of thousands of hours.
- Suggests increasing overall system availability by providing h/w spares…. an expensive approach!

Consider

- Historically, s/w improvements continuously reduce the amount of h/w required to enable capability.
- E.g., the TCP/IP capability in the early ARPANET nodes, which required two to four PDP-10/11 then, is now handled by a few thousand lines of code in a Windows or MAC machine.

Objective

- Do not dilute the strong h/w availability we've already captured
- Capture a similar process for software.
  - Employ objective measures like we already have for h/w
  - Use objective measures as thresholds and objectives for deliverables.


Information Availability
1.00000
Capability Lifecycle
IOC
H/W MTBF
New Increment
Observed H/W
Development Cycles
S/W Trouble Reports
H/W Redundancy requirement
New Increment
COTS S/W upgrades
Observed S/W
Development Cycles

Proposed Methodology

- Base RAM metrics on "Sustainability KPP" = A_{nr}
- Use Moore's Law 18 month technology refresh time line as the delivery cycle for transferring increments of functionality.
- Establish Business Process Model (BPM) as the requirement set.
- Set the threshold and objective RAM targets inside the 18 month delivery cycle.
- Adjust RAM targets for each successively delivered COTS s/w bundle to anticipate inevitable reduction in h/w redundancy requirements.

Engineering Tasks

- Isolate the BPM against the enterprise capability requirements.
- Establish the COTS s/w trajectory for this technical capability (COTS supportability).
- Establish threshold and objective RAM targets for the bundled s/w.
- Make RAM targets part of the IOC deliverables.